USP - ICMC - SSC

SSC 0511 - Sist. Informação - 2o. Semestre 2014

Disciplina de Organização de Computadores Digitais

Prof. Fernando Santos Osório

Email: fosorio [at] { icmc. usp. br , gmail. com }

Página Pessoal: http://www.icmc.usp.br/~fosorio/

Material on-line: Wiki ICMC:

http://wiki.icmc.usp.br/index.php/SSC-511-2014(fosorio)

IRM
Lab. de Robótica Móvel
CROB
Centro de Robótica da USP

Aula 05s

Apresentação da Aula

Conteúdos Abordados:

1. **Microprocessador 6502**

   - Simulador do 6502
   - Programação em Assembly 6502

Microprocessador 6502

Programação 6502 – Instruction Set

Principais Instruções

Load and Store Group

●	LDA	Load Accumulator	N,Z
	LDX	Load X Register	N,Z
	LDY	Load Y Register	N,Z
●	STA	Store Accumulator	
	STX	Store X Register	
	STY	Store Y Register	

Arithmetic Group

●	ADC	Add with Carry	N,V,Z,C
●	SBC	Subtract with Carry	N,V,Z,C

Increment and Decrement Group

●	INC	Increment a memory location	N,Z
	INX	Increment the X register	N,Z
	INY	Increment the Y register	N,Z
●	DEC	Decrement a memory location	N,Z
	DEX	Decrement the X register	N,Z
	DEY	Decrement the Y register	N,Z

Register Transfer Group

TAX	Transfer accumulator to X	N,Z
TAY	Transfer accumulator to Y	N,Z
TXA	Transfer X to accumulator	N,Z
TYA	Transfer Y to accumulator	N,Z

Logical Group

AND	Logical AND	N,Z
EOR	Exclusive OR	N,Z
ORA	Logical Inclusive OR	N,Z

Compare and Bit Test Group

●	CMP	Compare accumulator	N,Z,C
	CPX	Compare X register	N,Z,C
	CPY	Compare Y register	N,Z,C
	BIT	Bit Test	N,V,Z

Shift and Rotate Group

ASL	Arithmetic Shift Left	N,Z,C
LSR	Logical Shift Right	N,Z,C
ROL	Rotate Left	N,Z,C
ROR	Rotate Right	N,Z,C

* Simulador: 6502 Simulator / Site: http://home.pacbell.net/michal_k/ / By Michal Kowalski

Microprocessador 6502

Programação 6502

Instruction Set

Principais Instruções

Subroutine and Interrupt Group

● JSR	Jump to a subroutine	
● RTS	Return from subroutine	
BRK	Force an interrupt	B
RTI	Return from Interrupt	All
NOP	No Operation	

Jump and Branch Group

● JMP	Jump to another location	
BCC	Branch if carry flag clear	
BCS	Branch if carry flag set	
● BEQ	Branch if zero flag set	
BMI	Branch if negative flag set	
● BNE	Branch if zero flag clear	
BPL	Branch if negative flag clear	
BVC	Branch if overflow flag clear	
BVS	Branch if overflow flag set	

Stack Group

TSX	Transfer stack pointer to X	N,Z
TXS	Transfer X to stack pointer	
● PHA	Push accumulator on stack	
PHP	Push processor status on stack	
● PLA	Pull accumulator from stack	N,Z
PLP	Pull processor status from stack	All

Status Flag Change Group

CLC	Clear carry flag	C
CLD	Clear decimal mode flag	D
CLI	Clear interrupt disable flag	I
CLV	Clear overflow flag	V
SEC	Set carry flag	C
SED	Set decimal mode flag	D
SEI	Set interrupt disable flag	I

* Simulador: 6502 Simulator / Site: http://home.pacbell.net/michal_k/ / By Michal Kowalski

Microprocessador 6502

Programação 6502 – Addressing Modes

Principais Modos de endereçamento

Absolute (Direct) addressing

The operand is the address in memory where the data item can be found.

LDA	$6F55		This instruction will access memory location $6F55, and copy the contents into the accumulator register.

Immediate addressing

The operand is the data item.

LDA	#41		This instruction will load the number 41 ($29) into the accumulator register.

* Simulador: 6502 Simulator / Site: http://home.pacbell.net/michal_k/ / By Michal Kowalski

Microprocessador 6502

Programação 6502

Principais Modos de endereçamento

Indexed addressing

The operand is added to the contents of the index register, and this gives the location in memory of the data.

LDA	$4F10,X		$34		This instruction will access the memory location $4F44, and copy the contents into the accumulator.
			Index Register X		

Indirect addressing

The operand is the address of a memory location which contains the address of the data item.

LDA	($2B57)		2B57	3C
			2B58	6A
			6A3C	44

This instruction will load the data item 44 into the accumulator register.

* Simulador: 6502 Simulator / Site: http://home.pacbell.net/michal_k/ / By Michal Kowalski

Microprocessador 6502

Programação 6502

Principais Modos de endereçamento

Relative addressing

This is used in conjunction with branches. The operand is a 2's comp number (-128 - +127) which is added to the address of the instruction following the branch instruction to give the address of the instruction to be jumped to.

4545	BEQ	$13	This BEQ command will cause control to be passed to the instruction in location $4547 + $13 = $455A, provided that the zero flag is set.
4547	LDA	$9333	

Resumo: (ver documentação associada ao Simulador do 6502)

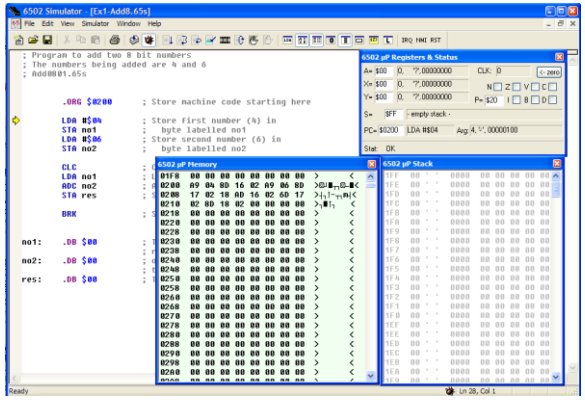
Absolute aaaa
Zero Page aa
Immediate #aa
Implicit
Accumulator A
Absolute Indexed, X aaaa, X
Absolute Indexed, Y aaaa, Y
Zero Page Indexed, X aa, X
Zero Page Indexed, Y aa, Y
Indirect Absolute (aaaa)
Indexed Indirect (aa, X)
Indirect Indexed (aa), Y
Relative aa

* Simulador: 6502 Simulator / Site: http://home.pacbell.net/michal_k/ / By Michal Kowalski

Entrada e Saída

Input/Output (I/O) – Entrada/Saída (E/S):

- Método usado para acessar os dispositivos de I/O no 6502



- I/O mapeado em memória
  Acesso a memória de I/O
  Instruções tipo LDA, STA

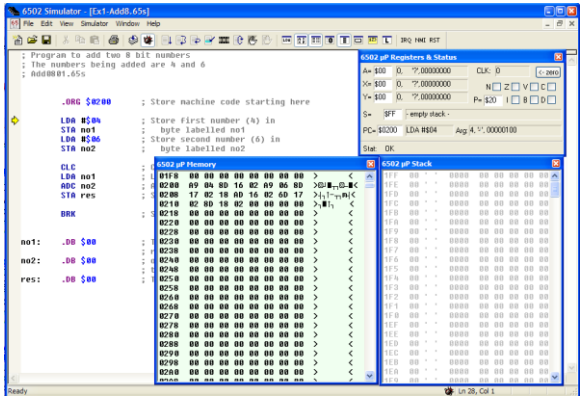
$E000 Clear terminal window
$E001 Ascii code will be sent to terminal screen and displayed as a character
$E002 Ascii code will be sent to screen and displayed as above, but CR/LF will be ignored
$E003 Ascii code will be sent to screen and displayed as a hexadecimal number

$E005 Controls horizontal location of cursor
$E006 Controls vertical location of cursor.

Entrada e Saída

Input/Output (I/O) – Entrada/Saída (E/S):

- Método usado para acessar os dispositivos de I/O no 6502



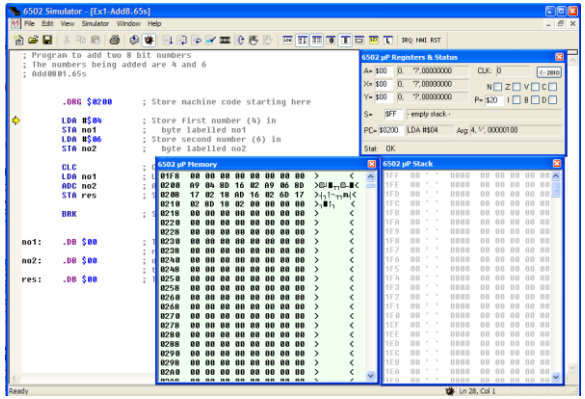
$E000 Clear terminal window
$E001 Ascii code will be sent to terminal screen and displayed as a character
$E005 Controls horizontal location of cursor
$E006 Controls vertical location of cursor.

```
io_putc    = $E001
io_posx    = $E005
io_posy    = $E006

           .ORG $300
inicio:    LDA #$0A
           STA io_posx                ; This will send cursor to column 10
           LDA #$05
           STA io_posy                ; This will send cursor to row 5
           LDA #'A'
           STA io_putc
           STA io_putc
           STA io_putc
           BRK
```

Input/Output (I/O) – Leitura de Teclado

- Método usado para acessar os dispositivos de I/O no 6502



- I/O mapeado em memória
  Acesso a memória de I/O
  Instruções tipo LDA, STA

$E004 If a character has been typed
into the terminal screen,
it will be copied to this location

Teclado...
ASCII, echo, Press-Release,
Leitura múltipla, Scan-Codes,

```
            .ORG  $1000
inicio:     LDA #$00
            STA $E000   ; Limpa Tela
Loop:       LDA $E004
            CMP #00     ; Espera tecla
            BEQ Loop
            STA $E002   ; Exibe Echo
Pula:       CMP #$0D    ; Enter: FIM
            BNE Loop
            BRK

; Leitura de teclado por "Pooling"
```

Entrada e Saída

Programando o 6502...

* Exemplos de Programas (ver arquivo anexo Aula05s)

- Leitura do Teclado (**LeTeclas.65s**)

- Programa de "Jogo Simples" (**Jogo01.65s, Jogo02.65s**)
  > Mover um caracter na tela (lê teclado, escreve na tela)

- Programa para procurar um Valor (**FindNum.65s**)
  > Uso de instruções em modo de endereçamento indexado

- Programa para exibir um "dump" de memória em Hexadecimal
  > Uso de endereçamento indexado (**DumMem.65s**)

- Programa para preencher a memória com um determinado valor
  > Uso de endereçamento indexado c/ + de 256 bytes (**FillMem.65s**)

INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO
Web institucional: http://www.icmc.usp.br/ssc/
Página pessoal: http://www.icmc.usp.br/~fosorio/
E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

Disciplina de Organização de Computadores Digitais / BSI
Web disciplina: Wiki ICMC - Http://wiki.icmc.usp.br
> Programa, Material de Aulas, Critérios de Avaliação,
> Lista de Exercícios, Trabalhos Práticos, Datas das Provas